etextools examples

Examples for some macros provided by the etextools package

<FC – April 20, 2010>

Contents

1 \expandnext examples **1**
1.1 Test if the replacement text of macro is really empty . . . 1
1.2 Test if the replacement text of a macro is blank (empty or spaces) . . . 2

2 \ExpandNext examples **2**
2.1 Test the parameter string of a macro . . . 2

3 Testing characters **3**
3.1 \ifsinglechar versus \iffirstchar . . . 3
3.2 Fully Expandable starred macros . . . 3
3.3 Fully Expandable macros with options . . . 3

4 Lists management **4**
4.1 \csvloop and \csvloop* examples . . . 4
4.1.1 \makequotes . . . 4
4.1.2 \detokenize . . . 4
4.1.3 \numexpr . . . 4
4.1.4 protected \textbf . . . 4
4.2 Index in lists and items by index . . . 4
4.2.1 \listloop: getting specific item . . . 4
4.2.2 \getlistindex . . . 5
4.2.3 \gettokslistindex with \ifcase . . . 5

1 \expandnext examples

1.1 Test if the replacement text of macro is really empty

```
\def\xx{ }
⟶ \edef \expandnextTest {\string \xx \ is \expandnext \ifempty {\xx }{}{not} empty}
\expandnextTest=  macro:->\xx\ is not empty
```

```
\def\xx{}
⟶ \edef \expandnextTest {\string \xx \ is \expandnext \ifempty {\xx }{}{not} empty}
\expandnextTest=  macro:->\xx\ is  empty
```

1.2 Test if the replacement text of a macro is blank (empty or spaces)

```
\def\xx{something}
⟶ \edef \expandnextTest {\string \xx \ is \expandnext \ifblank {\xx }{}{not} blank}
\expandnextTest=  macro:->\xx\ is not blank
```

```
\def\xx{ }
⟶ \edef \expandnextTest {\string \xx \ is \expandnext \ifblank {\xx }{}{not} blank}
\expandnextTest=  macro:->\xx\ is  blank
```

2 \ExpandNext examples

Example of the main documentation file to reverse the order of the characters in a string:

```
def\swap#1#2{{#2#1}}   \def\do[#1]#2{\swap #2}
edef\result{\naturalloop[\do]{4}{12345}}
meaning\result = macro:->{21}345{{21}345{{21}345{{21}345{12345}}}}
ExpandNext{\def\RESULT}{\naturalloop[\do]{4}{12345}}
meaning\RESULT = macro:->{21}345{{21}345{{21}345{{21}345{12345}}}}
```

2.1 Test the parameter string of a macro

The following commands create the filter for the string: "[#1]#2"":

```
ExpandNext{\DeclareStringFilter\ParaFilt}
     {\ExpandAftercmds\@gobblescape{\expandafter\string\csname[#1]#2\endcsname}}
```

1) \csname[#1]#2\endcsname is expanded first
2) Immediately after: \string
3) At this stage: \[#1]#2 (everything in category code other) is no more expandable
4) Then \ExpandAftercmds expands \@gobblescape
5) [#1]#2 is no more expandable
6) Then \ExpandNext expands its first argument: \DeclareStringFilter\ParaFilt{[#1]#2}

Remark: \detokenize would have doubled the # characters. Another possibility is to temporarily change the category code of # to 12 (other):

```
begingroup\catcode`\#=12
 \DeclareStringFilter\ParaFilt{[#1]#2}       global declaration
endgroup
```

```
\def\macroA#1#2{Something to do with #1 and #2}
\def\macroB[#1]#2{Something to do with #1 and #2}
\ExpandNext{\ParaFilt=.}{\parameters@meaning\macroA}{macro complies with [\#1]\#2}
                                                    {macro does not comply }
                                                          macro does not comply
\ExpandNext{\ParaFilt=.}{\parameters@meaning\macroB}{macro complies with [\#1]\#2}
                                                    {macro does not comply }
                                                          macro complies with [#1]#2
```

3 Testing characters

3.1 \ifsinglechar versus \iffirstchar

```
⟶ \edef \ifsinglecharTest {\ifsinglechar *{*hello*}{ single star }{ something else }}
  \ifsinglecharTest=  macro:-> something else
```

```
⟶ \edef \ifsinglecharTest {\ifsinglechar *{ *}{ single star }{ something else }}
  \ifsinglecharTest=  macro:-> something else
```

```
⟶ \edef \ifsinglecharTest {\ifsinglechar *{ * }{ single star }{ something else }}
  \ifsinglecharTest=  macro:-> something else
```

Note the space **after** the star ↑.

```
⟶ \edef \iffirstcharTest {\iffirstchar *{*hello*}{ first char is star }{ something else }}
  \iffirstcharTest=  macro:-> first char is star
```

3.2 Fully Expandable starred macros

```
\def\starmacro#1{\FE@ifstar {#1}\starred \notstarred }
\def\starred#1{your "#1" will be processed by the STAR form}
\def\notstarred#1{your "#1" will be processed by the NORMAL form}
⟶ \edef \FE@ifstarTest {\starmacro {sample text}}
  \FE@ifstarTest=  macro:->your "sample text" will be processed by the NORMAL form
```

```
⟶ \edef \FE@ifstarTest {\starmacro *{sample text}}
  \FE@ifstarTest=  macro:->your "sample text" will be processed by the STAR form
```

3.3 Fully Expandable macros with options

```
\def\optmacro#1{\FE@testopt {#1}\OPTmacro {Mr.}}
\def\OPTmacro[#1]#2{#1 #2}
⟶ \edef \FE@testoptTest {\optmacro {Woody Allen}}
  \FE@testoptTest=  macro:->Mr.  Woody Allen
```

```
⟶ \edef \FE@testoptTest {\optmacro [Ms.]{Vanessa Paradis}}
  \FE@testoptTest=  macro:->Ms.  Vanessa Paradis
```

4 Lists management

4.1 \csvloop and \csvloop* examples

4.1.1 \makequotes

```
\def\makequotes#1{"#1"\space }
⟶ \edef \csvloopTest {\csvloop *[\makequotes ]{hello,world}}
\csvloopTest=  macro:->"hello" "world"
```

4.1.2 \detokenize

```
⟶ \edef \csvloopTest {\csvloop *[\detokenize ]{\un ,\deux }}
\csvloopTest=  macro:->\un \deux
```

4.1.3 \numexpr

```
\def\mylist{1,2,3,4,5}
\def\BySeven#1{$#1\times 7 = \number \numexpr #1*7\relax $\par }
⟶ \edef \csvloopTest {\csvloop [\BySeven ]\mylist }
\csvloopTest=  macro:->$1\times 7 = 7$\par $2\times 7 = 14$\par $3\times 7 = 21$\par
               $4\times 7 = 28$\par $5\times 7 = 35$\par
```

4.1.4 protected \textbf

```
⟶ \protected@edef \csvloopTest {\csvloop *[\textbf ]{hello ,my ,friends}}
\csvloopTest=  macro:->\protect \textbf  {hello }\protect \textbf  {my }\protect \textbf
               {friends}
```

4.2 Index in lists and items by index

4.2.1 \listloop: getting specific item

```
\csvtolist*[\mylist]{one,two,three,four,five,alpha,beta,gamma}
⟶ \edef \getlistitemTest {\listloop [4]\mylist }
\getlistitemTest=  macro:->five
```

4.2.2 \getlistindex

```
⟶ \getlistindex[\myindex]{alpha}\mylist
  \myindex=  macro:->5
```

```
  \newcount\myindex
⟶ \getcsvlistindex*[\myindex]{alpha}{one,two,three,four,five,alpha,beta}
  \the\myindex=    5
```

4.2.3 \gettokslistindex with \ifcase

Always purely expandable (no need of \pdfstrcmp, comparison done by \ifx):

```
⟶   \ifcase  \gettokslistindex{D}{LRDF\relax O}
          What do to if L
     \or  What do to if R
     \or  What do to if D
     \or  What do to if F
     \or  What do to if \relax
     \or  What do to if O
     \else Problem
     \fi
Result=    What do to if D
```